

Strategic Web Based Management Game

Group 12

Per Eriksson

Simon Ragnar

Per Strand Eklund

Ingemar Markström

Max Walter

Project Overview Document

Who are the users and what problem does the system solve for them?

- For this project we are going to produce a web-based, massive multi player, strategy game.
- The users are people approximately between 12-30 who like to play web-based massive multi player strategy and/or role-playing games on PC or console.
- The main purpose of the game is to serve as entertainment over time. Having that said, it's not aimed towards people who are looking for a game to play over several hours in a row but rather as a continual, “in the back of their minds”, planning and plotting of strategies to be used in the game.
- The game will also, to a certain degree, provide the users with a social sphere that can become more or less important to the player depending on how long he/she has been playing the game and how much the player likes to interact with other players. For example, for some people the social aspect of the game can be crucial in order to keep them playing.

The main uses of the system

The main use of our web-game is for the player to have fun and feel like he/she has accomplished something when he stops playing. Because of the structure of the game, where the time spent on playing the game will be limited to short periods of time, the thought is that the player will be coming back to the game in order to become successful. Because of that the game will be running continuously on a server where the players progress will be saved. Players' actions can effect even offline players as the game will be constantly updated. The player will be able to come back to the game universe and continue on from where he left off, but the players around him will probably have made some moves too, and even made moves affecting the player. Because of the fact that you cannot play for more than a limited amount of time, in a sense that you will not have adequate resources to perform more than a few moves per day affecting the in-game world. However you can still interact with other players through forums, message boards etc. The main objective of the game is not necessarily to steal all of the player's concentration, but to give the him/her the possibility to play the game along with other games.

Example:

A player is waiting for an online multiplayer game to start. He glances at his watch, and realizes that he has the possibility to log in and make a move in our game. He opens up a web-browser, goes to the game-page, and comes to the login screen. After logging in, a new screen appears where he can either choose to manage his account or play the game. There he can see where he is on the play field, nearby enemies, friends, teammates and other relevant information in lists. The player decides to move his ship closer to a teammate and shoots an enemy with his slow-time weapon. He quits the game by logging out. He jumps back to the waiting screen in the online multiplayer game he was playing before and realizes that he has not lost anything but some waiting time.

An other example may be:

Some students are sitting in school studying and have been doing so for the last three, maybe four, hours. Suddenly they decide to take a short break and go buy something in the store. On the way to the store one of the students picks up his mobile phone, brings up the mobile web browser, and logs

on to the game web page. After performing a complex and rather daring strategic action he messages his teammates about the attack and then puts down his phone. One of the other students ask something about who he may have been messaging, and he answers that he was checking his progress in the game. After a relatively short description, the other student brings up his phone, and creates a new account. When back in school again they both log in, and discover that the attack went as planned. They as a team has gained some precious ranking points.

The context/environment in which the system is to be used

- Since our game will be played during short time intervals, it'll be used anywhere where you have a computer and Internet connection available. At home will be the most common place for this of course, but other places are for example at school or work during breaks, at a friends house or perhaps on your laptop when you travel.
- The game will be web based and played through a web-browser, hence no installation needs to be made.

The scope of the system

- Inside the scope
 - Log-in system
 - Massive Multi Player
 - Real-time interaction
 - Interaction between players (messaging, combat)
 - High-score lists
 - Alliances between users
 - Forums/message boards
 - Definite goal (when reached, the game ends and starts over)
 - Pictures
- Outside of the scope
 - Animation
 - Music/Sound effects

The main factors that need to be taken in to account when designing and building the system

- Usability
 - For example, there shall be an easy to read and understand instructions-section on the web-page. Here, users can easily look up how to do what they want to do. For example a user might want to create a new alliance, but he/she doesn't know how, so she goes to the instructions section and there she gets guided on how to create a new alliance.
- Intuitive
 - A clear and appealing graphic design of the web page considering our target group.
- Rewarding
 - By being able to upgrade their ship (for example engines, weapons, shields etc.) and advancing towards the goal of the game (both the point of where the game ends and also to be closer to this goal compared to other players). This makes people feel that they accomplish something which is rewarding in itself.
- Fun
 - The ability to interact and play against, and with, other people, attacking your enemies and helping your friends. Constant evolution of your spaceship, where you are in charge of what upgrades to make and what abilities you want to achieve.
- Keep users playing (if the playerbase drops to zero no one will talk about our game and no

new players start playing)

- We need to consider how to get an initial playerbase.
- Keep a flow of new players coming in.

It's important that we are aware of that it'll not be possible to keep a high player-base all the time. At some point it will steadily go down until the game is more or less dead. We need to take into account that most web-based games have their peak-period of popularity and can therefore be considered as "trendy".

Technologies and Risks

- Risks
 - We need to consider the security of the users and make sure that no one "hacks" into any user accounts and erases all they've achieved.
 - To fail in creating an appealing game
- Technologies (this might change as the project progresses)
 - Databases
 - AJAX
 - HTML
 - PHP
 - Javascript

6. Technologies that we plan to use and associated risks.

In this stage of the project we have come to the conclusion that we will use the following four programming languages: Javascript, AJAX, HTML, CSS, PHP and SQL. Due to the fact that none of the members in the group has ever done web applications on this scale before, there is a medium to high risk that one or more of these languages will be changed to some other language. These are the decided languages with the following motivations as to why we've chosen them:

Javascript and AJAX:

Javascript and AJAX are both web development languages. They are mostly used for client side aspects of a page that cannot be written in HTML, for instance, behind the scenes, application code. AJAX stand for Asynchronous Javascript And XML and as the name implies uses Javascript. The difference is that with AJAX, you can create real time updates to the web-page without having to manually update the page. This will be useful for a diverse set of development problems in the game. Imagine for example the basic concept that you would like some variable to be displayed in real-time, then AJAX solves that problem.

Risks with Javascript and AJAX:

Contrary to what you may believe, Javascript is not based on the programming language Java but is rather designed to have a similar look to Java. What this means remains to be discovered as we've not had any experiences with this language or with AJAX, but if problems should arise, the probability that they are threatening to the project is most likely small.

HTML and CSS:

These languages will be used to solve the problem of structuring and designing the web-page that the game is based around. Basically, everything that the user sees in the web browser will be written in HTML or CSS. CSS is short for Cascading Style Sheets and its practical use is to describe the presentation of a document written in a markup language, that is in our case, HTML. In other words, the game web page will be written in HTML and CSS will be used to make the web page look better.

Risks with HTML and CSS:

The main concern here is CSS. The group suffers from lack of knowledge in any extensive practical use of the language. What this means is that there is an estimated medium to high risk that a big portion of time will have to be spent on learning how to use these languages to make a satisfactory layout and structure of the web-page. This, and the use of HTML is one of the big concerns of the project. However, complete failure in this field is estimated as low. This is because of the abundant amount of information on the Internet concerning these two languages.

SQL:

The project will have use of a database. All players will have an arbitrary amount of information stored about their accounts individually. We will solve this problem by the use of MySQL which is a database language.

Risks with SQL:

All members of the group have a firm understanding of large aspects of this problem, from the design of a database, to the implementation of a dialog between database and application. The problem that could arise here is how database programming (connecting the database to the web-page) with PHP is implemented. Even though there are large amounts of information about this on the Internet, there are still risks involved. For example, it might take more time than we plan.

PHP:

This leaves us with the server aspect of the project which will be solved by using PHP. PHP is a general purpose scripting language but has its principal focus on server-side scripting. In short, its practical use will be conversing with the database and delivering web-pages on demand.

Risks with PHP:

Specific members of the group have a good understanding in use of the language. Therefore there's an estimated low risk involved in using this particular language to solve the problem. However, due to the nature of the problem being solved, that is the bulk of the server scripting, a preliminary prediction is that this might develop into one of the biggest parts of the project.

Overall risks:

The group perceives the problem as a fairly simple, however we must take into account all technologies involved. In other words, the problem the group is facing is not in the actual coding itself. We are all fairly good at implementing advanced algorithms in different languages. What we are lacking is a broad know-how of the mentioned technologies/languages above and exactly how they will fill their role in the project. That is, how do you implement a massive multi player web based game? To be certain, no advanced algorithms will be used, but rather a complex network of simple functions and concepts yet to be discovered.